Global Modulation Versus Local Activation Functions in Neural Systems

Conventional neural networks are usually described as alternating between linear layers and local nonlinear activation functions such as ReLU, sigmoid, or tanh. In this picture, each neuron independently applies a scalar nonlinearity to its own activation value.

This viewpoint is historically successful, but it can also obscure a more general interpretation of what makes these systems expressive.

A stack of fixed linear transformations collapses into one affine operator:

$$y = BAx = Mx$$

so some mechanism is needed to prevent this affine collapse. Local activation functions achieve this indirectly by making neuron participation data dependent. A ReLU neuron either passes or suppresses its channel depending on its local activation value.

However, there is no fundamental requirement that this conditional behavior must occur only through local scalar nonlinearities.

An alternative viewpoint is to treat the network as a dynamically modulated operator system. Instead of attaching nonlinear functions to individual neurons, one allows the transformation itself to vary according to the input, state, or context.

For example:

$$y = (W \odot G(x))x$$

where:

- W is a base weight matrix,
- $G(x)$ is a data-dependent modulation field,
- $\odot$ denotes elementwise multiplication.

The important conceptual shift is that computation is no longer viewed as:

> fixed weights plus local nonlinearities

but rather as:

dynamically reconfigured operators.

In this framework, weights acquire conditional relevance. For one input, certain pathways may be strongly emphasized, while others are attenuated or even sign-inverted. Parameters no longer have fixed global roles; their participation depends on the current computational context.

This perspective can be justified naturally from the standpoint of affine collapse. Once the effective operator depends on the signal:

$$M(x)$$

the system can no longer be reduced to one fixed affine map. Expressive power comes from dynamic operator variation rather than merely from composing static operators with scalar nonlinearities.

Importantly, modulation need not be binary. ReLU-style gating corresponds roughly to gains in $\{0,1\}$, but continuous modulation allows:

- partial participation,
- amplification,
- suppression,
- sign inversion,
- interference effects,
- smooth operator morphing across input space.

This opens a much broader design space than the traditional neuron-centric perspective.

Modulation can operate on:

- individual weights,
- rows or columns,
- channels,
- tensor blocks,
- latent factors,
- spectral components,
- attention pathways,
- or entire operators.

Similarly, modulation signals can depend on:

- the current input,
- recurrent state,

- memory,
- time,
- control systems,
- external context,
- or learned routing mechanisms.

The resulting systems begin to resemble adaptive physical processes more than static feedforward circuits. Computation becomes a process of dynamically shaping transformation geometry rather than simply applying fixed layers separated by pointwise nonlinearities.

This viewpoint also unifies many seemingly different architectures. ReLU gating, attention, mixture-of-experts systems, adaptive filters, hypernetworks, and dynamically modulated transforms can all be interpreted as methods for constructing context-dependent operators that avoid affine collapse through conditional parameter participation.